In computing and engineering, much has been written lately about the glass being half empty – the increasing need for science and engineering, the difficulty of attracting students to these demanding disciplines, the gender gap, H1B visa quotas, off-shoring, the flat world, and so on. From where I sit, the glass looks way more than half full – we cannot supply the demand for computer scientists; our undergraduate students are starting out making $50-70K; they are working in interesting jobs; the technologies they develop change the world for the better; and we can predict an increasing need for innovation in computing for years to come.

Against the backdrop of the Knowledge Cycle architecture (see companion column), the rest of this article identifies myths or simplistic beliefs where a better understanding of a reality can lead to a significant benefit. Some of the myths as stated are red herrings in that educators constantly work toward the realities but my own observation is that the myths do represent attitudes or inertia and the realities represent enriched experiences or at least alternative views worth considering.

**Myth #1:** It doesn’t matter what you major in, just get your college degree.

**Reality:** Getting your degree *is* important and liking what you do is *really* important but what you major in can make the difference between stepping directly into a career from the minute you graduate or becoming part of the 65% of the boomerang kids who go home to live with their helicopter parents after college graduation (Money Magazine, Dec 29, 2006). No college degree yields minimum wage jobs paying around $15K a year. At the bachelors level, in salaries, the kind of degree can make the difference between making less than $30K or more than $60K upon graduation and a bigger difference later on (see companion table showing salaries by degree). It would take around 24 years for someone who starts at $30K to make $60K if they received an average raise of 3% a year (and their real earning power is closer to zero if the annual cost of living increase is, say, 3%!).

[Note: Parents owe it to their kids to give them the gift of independence - starting with allowances, chores, summer camps, summer jobs, ... with a strong presupposition that the kid will be on their own when they finish college. In teaching the computing Capstone course at University of Arkansas, I tell college seniors that they are not allowed to go home and live with their parents after college. Then I proceed to tell them they have their senior year to plan what happens next. See Myth 18.]

**Myth #2:** Money isn’t the most important consideration.

**Reality:** Of course that is true too, but it’s less fun living with less income. Much more importantly, hard work, responsibility, opportunity, and enriching careers tend to correlate with money. So, think of this differently, do you want an interesting life with lots of challenge? Then work hard from day one (in your chosen career whatever it is). Majoring as an undergraduate in technical disciplines often gives you a significant edge if you later go into business, law, or continue into the workforce. One other slant on money involves student loans - *Students*, do whatever you can to minimize taking loans since racking up loans beyond $30K can take years to
pay off. Better, get a great job during summers or even school to help pay these back as soon as possible. Don’t just assume that because others are getting loans, it is the only way to go.

Myth #3: When you turn on the hard-work after burners doesn’t matter – spend your freshman year getting used to college.

Reality: Assuming a student has prepared well in high school, the most critical advising point is at the beginning of freshman year. Even most well-prepared high school seniors and their generally clueless parents will not realize that, if there is ever a time to target math, science, or engineering as an major, it is first semester freshman year and students do this by taking Calculus, a science like Physics, freshman English and the university level course in a technical major they are interested in. If students miss this window, it is later much harder to move from liberal arts or business or other disciplines to technology whereas it is relatively easy for technically prepared students to move to these other disciplines.

Note that “being prepared in high school” would have required a student to take the usual range of college preparatory classes. So the work ethic begun in junior high and high school pays back in college and in life. In fact, going back to the beginnings, the best things I did for and with my children were reading books when they were young (6 months to 12 years old), practicing the addition and multiplication tables with timed tests until they were faster than I was, and always being interested and beamingly proud of their academic, extracurricular, and sports accomplishments.

Myth #4: Undergraduate advising should be focused mainly on what classes students should take.

Reality: When I returned to teach at a university after a career in industry, I was shocked to learn that colleges usually measure the six-year graduation rate for undergraduates. So, I agree that competent undergraduate course advising should help students progress through school at a reasonable pace, in four or sometimes five years. But advising and courses should also provide students with an understanding of the scope of their discipline (for instance, computing), the wide range of interesting applications, how it relates to other disciplines, and how it impacts society. Campus culture and foreign travel should be encouraged – you will remember every day of your trip overseas for the rest of your life. I spent a summer bicycling in England when I was 16 and my daughter and her friends hiked the 190 mile Coast-to-Coast Trail across northern England when she was a college junior. At the same time, interning, cooping and work experience related to the major should be strongly encouraged while still in school. Prospective employers will look at the whole person, not narrowly just at their grades. Those students who intern in their major from the summer following their sophomore or junior year on considerably enhance their job prospects. Some companies will not hire students who do not have interning experience.

Myth #5: Undergraduates should take classes and read text books.

Reality: While books do encapsulate a core of what is worth knowing and hint at the shape of a field, many undergraduates get all the way through four years of school without ever reading a research paper, knowing about ACM and IEEE online, or attending a conference. So they do not know about the iceberg of rich resources they could use to solve problems or the excitement of brand new knowledge. Even in a subdiscipline of computer science like database management,
every day somewhere there is a conference going adding new knowledge that may not make it into textbooks for years. If you don’t believe me, browse the dbworld website.

Myth #6: Computer science undergraduates mainly learn how to program.

Reality: While it is true that computer science students should know how to program, they should gain many additional skills. I spend much of my time reading, writing or presenting. Our industrial advisory board faults all of engineering for turning out students who know too little about business so we are working on ways to inject business knowledge back into our programs.

Myth #7: It is OK to teach software engineering without teaching computational tools. In general, it is better to teach undergraduates concepts rather than skills.

Reality: While it is true that students will be able to pick up tools and courses should not teach to skills, nevertheless, skills should be included in courses. Amazingly few faculty or students know how to use Microsoft Word effectively (i.e. features like document map, styles, revision mode). Some do not know what firewalls do or how to backup their machines. Tools like CVS, Javadocs, project management, IDEs, UML, XML, SOAP, … are not always part of courses but some of these should be.

Myth #8: Undergraduate students should not take graduate classes.

Reality: Undergraduates should be encouraged to take graduate classes. This is a bridge for encouraging undergraduates into graduate school. Surprisingly, many undergraduates have never considered graduate school and don’t know much about it until a faculty member explains the mechanisms for applying and the benefits. For many computing students, the best terminal degree is the Masters which requires two additional years past the bachelors.

Myth #9: A senior-level Capstone projects course is enough to teach undergraduates about projects. Teams, projects, and reinforcing software engineering and presentation skills should be the capstone focus.

Reality: While good, this is not enough. Project courses should start at the junior level. There is a need to teach career skills, how to start a small business, how to bid a contract, how to apply for a patent, etc. This can be part of Capstone or injected into other courses.

Myth #10: All student projects are toy world projects.

Reality: Students projects can matter, can lead to results that industry uses, can lead to inventions, SBIR proposals, business plans, small businesses, and industry standards. It is empowering to believe this and to make it happen. Students who have these experiences are sought after. My students have placed in national programming and business plan competitions, developed open source software and software for widespread release within companies, demonstrated to industry, executed contracts, started businesses, and presented papers at academic conferences.

Myth #11: Research is for graduate students.
**Reality:** In industry, people are asked to solve problems. So undergraduates need to learn how to frame problems as well as solve them. Also, in industry, problems do not have well defined scope unless you learn how to divide and conquer, that is, break big problems into smaller ones. Mixing undergraduates and graduates in problem solving and research is good for everyone. It has helped my undergraduates get accepted at Harvard, Stanford, Carnegie Mellon and other top schools or to get excellent jobs in industry (architect, business intelligence, ...). As an addendum to this myth, one reason some students shy away from research is that they have to write a thesis (at the Bachelors, Masters or PhD level). But it turns out that writing a thesis is not so hard if you just view it as filling in a form - telling the reader the title, the problem you are solving, the objective of the thesis, the related work you are building on, the approach you took to solving the problem, how you measured your success, your conclusions and lessons learned, and future work to continue to explore the question. A thesis template document is here that makes filling in the form easier.

**Myth #12:** We have succeeded when our students graduate.  
**Reality:** We have succeeded if our students graduate and already have a job or graduate school lined up. When I first started teaching I asked more than a few students, “So, graduating next week – what’s next?” The reply: “Oh, I guess I’ll get a job.” That is not the right answer! A metric of success should be, how many graduates with a job offer in hand. Seniors should start the process of applying to graduate schools in October or November of their senior years. Those going into the workforce should plan for a lighter course load their last semester; they should research employers, go to career fairs, and line up interviews. Surprisingly, many students do not know to do this unless you tell them and expect them then follow through.

**Myth #13:** A Bachelors degree is a good place to stop – it's time to get a job.  
**Reality:** Although not for everyone, I believe that for many, the M.S. degree should be the terminal professional degree for many computer science students. We should strongly encourage our undergraduates to think in terms of careers, not jobs, and M.S. rather than B.S. That said, our students might also go on productively in business or law or many other areas because computing

**Myth #14:** Grad school pays off.  
**Reality:** In fact, it does, but not necessarily in the way you might think. A masters degree (M.S., M.A.) generally takes 1.5-2 years while a doctorate (Ph.D.) takes an additional 3 or so years. A naïve expectation might be that, in the workplace, you might receive more respect and better work assignments from the higher degree. In fact, your co-workers might never know you have additional degrees. It is true that some employers might value a graduate degree as worth a higher starting salary, but once in the workplace, the metric of success is worker performance and impact – what did you do for me lately? Higher level degrees do not automatically buy this. What a masters can provide is significantly more depth and breadth of experience in the field than a Bachelors degree can – roughly doubling the range of in-depth courses as well as adding a research, problem solving component that generally ends in a masters thesis, which is a 30-60 page document that requires formal defense. Most of the time, the students who have this additional depth, breadth, experience, and technical maturity will compete better in the marketplace than they would have without the degree. The Ph.D. degree is a step beyond, often
preparing a student for a highly competitive life in research and teaching. For a time, the Ph.D. student is the world’s expert in a narrow area. If Ph.D. candidates only write a dissertation and do not gain other experiences in graduate school, then they may not have acquired the survival skills they will need. Ph.D. students be armed with survival skills – they should publish research papers, co-author proposals, participate in standards groups as major contributors, develop open source software and have other experiences that contribute to their professional maturity. If they go into academics, they will need these skills to compete with the best of the best for grant money and tenure (which is essentially a permanent position at their university). To return to the question, does grad school pay off? Statistically, yes, somewhat, perhaps not as much as expected (maybe 15-20% which is probably better than industry raises), but students who choose this route often have more stable positions and have a competitive edge.

**Myth #15:** Isolation is good - professors should not collaborate with each other and certainly should not with those in other departments or colleges.

**Reality:** When students collaborate with professors (plural) in joint projects, everybody wins. More than this, computing is by nature a tool that can be used in many disciplines. I went into computing so, like Peter Pan, I would not have to grow up and choose some specific discipline but instead could get involved in many areas: supply chain, healthcare, military, law, geophysics, paleoinformatics, …. This often means collaborating and crossing boundaries between disciplines.

**Myth #16:** Companies see universities as a good place to recruit new employees.

**Reality:** This “myth” is true. It should not be the whole story but often is - that is, too often, industry does not take much advantage of university research. Many companies would benefit if they saw university research as a gold mine. For the price of a graduate student and a few weeks of summer pay for a faculty member, they can contract with the university to explore areas that cost too much to explore in house. Benefits of funding a university team are that the professor and student(s) work on a real industry problem with an industry project champion. A direct benefit is that often significant progress is made toward solving the problem and the students are immediate candidates for hire, already trained. Collateral benefits are that many other students learn about the work, are interested, and find ways to get involved, raising that company’s profile in an increasingly competitive job market.

**Myth #17:** Faculty teach, students learn.

**Reality:** Faculty, you should look for ways to use your students to explore a new area, then they teach you. Faculty can use special projects classes to explore areas they want to know more about. Students get experience through this role reversal; faculty stay up to date.

**Myth #18:** It’s OK to go home to live with your parents.

**Reality:** No, it is not! We all want our children to become independent. Plan ahead. Make expectations clear even when students are freshman that they will be on their own financially when they graduate. It is a great gift (and not a right) if parents can support their students through a four year college education so that they start their financial lives debt free. After that, it is realistic that they should earn their own way. If students planned ahead in lining up a job,
they can transition to being self supporting. If they go on to graduate school, they should plan
ahead to apply early (in November or December) for teaching and research positions or campus
or intern positions while they go to school (starting the following August or September). They
should intern in the summers.

**Myth #19: Life happens to you.**

As my Mother used to tell me, “Be the head and not the tail,” that is, be the master of your own
destiny. You (not your parents, teachers or friends) are in charge of making your life as rich and
full as it can be -- life does not just happen to you if you choose to think ahead and use your
brain. You are much more likely to hit the target if you aim your arrows in that direction.

* * * * *

**Other “myths”**

Sure, there are other myths we could consider. Others have said much more about these topics
than I can in this column.

- there is off-shoring, but many companies we talk to strongly prefer to hire at home.
- there is a drop in enrollment in computing that seemed to lag the burst of the Internet
  bubble but computer science students still earn some of the most competitive salaries.
- there are fewer females in our workforce, perhaps because of perceptions of nerdiness or
  the isolation of late night solitary programming, but the reality is that most computer
  scientists work in teams and collaborate across discipline boundaries, many never
  program but instead analyze problems, and have considerable flex time for diverse
  pursuits; parents can help their daughters succeed in technical areas through
  encouragement, good role models, and the opportunity to engineer technical innovations
  that benefit society.

There are other mysteries I still don’t yet understand about colleges and computing. For
instance, why is it that my students stay the same age while I keep getting older? And how is it
that the supposedly quiet life of a university professor with three months off in the summer has
me running in more interesting directions than ever before? At a deeper level, I list one last
myth/mystery that will considerably affect undergraduate education in computing over the
coming years.

**Myth #20: Classical computer science is unchanging and our undergraduate program needs to
hug the traditional core.**

**Reality:** This is a hard one for me. It was interesting to me to come back to university teaching
after 23 years (several Internet lifetimes) and discover that core courses remained much the
same, e.g., operating systems, database management, formal languages, programming
languages, artificial intelligence. While newer programming languages like Java (no longer so
new!) have replaced Fortran and Pascal, the number of areas of computing to be interested in has
exploded. We have grids, games, robots, nano, bio, and many other new playgrounds. When I
was born, we had one ENIAC computer that weighed 27 tons; now every freshman has a more
powerful cell phone and companies like Microsoft and Intel are viewing cell phones as the
affordable computational Internet platform leapfrogging PCs and laptops in the rest of the world.
As we progress to smaller, more powerful, more plentiful computing, in even 20-30 more years,
every person will effortlessly manage thousands to millions of tiny computers (smart dust). Every other field will be absorbing computation that is progressively easier to use. So there is the ongoing challenge to evolve what we teach. However our undergraduate computing programs evolve, the field of computer science will continue to produce innovations that create jobs and benefit and change society.